

A Comparative Study of Different Reconstruction Schemes for a Reconstructed Discontinuous Galerkin Method on Arbitrary Grids

AIAA CFD Conference

Hong Luo
Hanping Xiao
Robert Nourgaliev
Chunpei Cai

June 2011

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint should not be cited or reproduced without permission of the author. This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights. The views expressed in this paper are not necessarily those of the United States Government or the sponsoring agency.

The INL is a
U.S. Department of Energy
National Laboratory
operated by
Battelle Energy Alliance



A Comparative Study of Different Reconstruction Schemes for a Reconstructed Discontinuous Galerkin Method on Arbitrary Grids

Hong Luo¹ and Hanping Xiao²

North Carolina State University, Raleigh, NC, 27695

Robert Nourgaliev³

Idaho National Laboratory, Idaho Falls, ID, 83415

Chunpei Cai⁴

New Mexico State University, Las Cruces, NM, 88001

A comparative study of different reconstruction schemes for a reconstruction-based discontinuous Galerkin, termed RDG(P1P2) method is performed for compressible flow problems on arbitrary grids. The RDG method is designed to enhance the accuracy of the discontinuous Galerkin method by increasing the order of the underlying polynomial solution via a reconstruction scheme commonly used in the finite volume method. Both Green-Gauss and least-squares reconstruction methods and a least-squares recovery method are implemented to obtain a quadratic polynomial representation of the underlying discontinuous Galerkin linear polynomial solution on each cell. These three reconstruction/recovery methods are compared for a variety of compressible flow problems on arbitrary meshes to assess their accuracy and robustness. The numerical results demonstrate that all three reconstruction methods can significantly improve the accuracy of the underlying second-order DG method, although the least-squares reconstruction method provides the best performance in terms of both accuracy and robustness.

I. Abstract

The discontinuous Galerkin methods¹⁻²⁵ (DGM) have recently become popular for the solution of systems of conservation laws. Nowadays, they are widely used in computational fluid dynamics, computational acoustics, and computational electromagnetics. The discontinuous Galerkin methods combine two advantageous features commonly associated to finite element and finite volume methods. As in classical finite element methods, accuracy is obtained by means of high-order polynomial approximation within an element rather than by wide stencils as in the case of finite volume methods. The physics of wave propagation is, however, accounted for by solving the Riemann problems that arise from the discontinuous representation of the solution at element interfaces. In this respect, the methods are therefore similar to finite volume methods. The discontinuous Galerkin methods have many attractive features: 1) They have several useful mathematical properties with respect to conservation, stability, and convergence; 2) The method can be easily extended to higher-order ($>2^{\text{nd}}$) approximation; 3) The methods are well suited for complex geometries since they can be applied on unstructured grids. In addition, the methods can also handle non-conforming elements, where the grids are allowed to have hanging nodes; 4) The methods are highly parallelizable, as they are compact and each element is independent. Since the elements are discontinuous, and the inter-element communications are minimal, domain decomposition can be efficiently employed. The compactness also allows for structured and simplified coding for the methods; 5) They can easily handle adaptive strategies, since

¹Associate Professor, Department of Mechanical and Aerospace Engineering, Senior Member AIAA.

²Ph.D Student, Department of Mechanical and Aerospace Engineering, Student Member AIAA.

³Senior Scientist, Multiphysics Methods Group, Member AIAA.

⁴Assistant Professor, Department of Mechanical and Aerospace Engineering, Senior Member AIAA.

refining or coarsening a grid can be achieved without considering the continuity restriction commonly associated with the conforming elements. The methods allow easy implementation of *hp*-refinement, for example, the order of accuracy, or shape, can vary from element to element; 6) They have the ability to compute low Mach number flow problems without recourse to the time-preconditioning techniques normally required for the finite volume methods. In contrast to the enormous advances in the theoretical and numerical analysis of the DGM, the development of a viable, attractive, competitive, and ultimately superior DG method over the more mature and well-established second order methods is relatively an untouched area. This is mainly due to the fact that the DGM have a number of weaknesses that have yet to be addressed, before they can be robustly used to flow problems of practical interest in a complex configuration environment. In particular, there are three most challenging and unresolved issues in the DGM: a) how to efficiently discretize diffusion terms required for the Navier-Stokes equations, b) how to effectively control spurious oscillations in the presence of strong discontinuities, and c) how to develop efficient time integration schemes for time accurate and steady-state solutions. Indeed, compared to the finite element methods and finite volume methods, the DG methods require solutions of systems of equations with more unknowns for the same grids. Consequently, these methods have been recognized as expensive in terms of both computational costs and storage requirements.

Dumbser et al.¹⁸⁻²⁰ have originally introduced a new family of reconstructed DG methods, termed PnPm schemes, where Pn indicates that a piecewise polynomial of degree of n is used to represent a DG solution, and Pm represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes. The beauty of PnPm schemes is that they provide a unified formulation for both finite volume and DG methods, and contain both classical finite volume and standard DG methods as two special cases of PnPm schemes, and thus allow for a direct efficiency comparison. When $n=0$, i.e. a piecewise constant polynomial is used to represent a numerical solution, P0Pm is nothing but classical high order finite volume schemes, where a polynomial solution of degree m ($m \geq 1$) is reconstructed from a piecewise constant solution. When $m=n$, the reconstruction reduces to the identity operator, and PnPn scheme yields a standard DG method.

Obviously, the construction of an accurate and efficient reconstruction operator is crucial to the success of the PnPm schemes. In Dumbser's work, this is achieved using a so-called in-cell recovery similar to the inter-cell recovery originally proposed by Van Leer et al., where recovered equations are obtained using a L_2 projection, i.e., the recovered polynomial solution is uniquely determined by making it indistinguishable from the underlying DG solutions in the contributing cells in the weak sense. The resultant over-determined system is then solved using a least-squares method that guarantees exact conservation, not only of the cell averages but also of all higher order moments in the reconstructed cell itself, such as slopes and curvatures. However, this conservative least-squares recovery approach is computationally expensive, as it involves both recovery of a polynomial solution of higher order and least-squares solution of the resulting over-determined system. Furthermore, the recovery might be problematic for a boundary cell, where the number of the face-neighboring cells might be not enough to provide the necessary information to recover a polynomial solution of a desired order.

Fortunately, recovery is not the only way to obtain a polynomial solution of higher order from the underlying discontinuous Galerkin solutions. Rather, reconstruction widely used in the finite volume methods provides an alternative, probably a better choice to obtain a higher-order polynomial representation. Luo et al. develop a reconstructed discontinuous Galerkin method using a Taylor basis³⁵⁻³⁷ for the solution of the compressible Euler and Navier-Stokes equations on arbitrary grids, where a higher order polynomial solution is reconstructed by use of a strong interpolation, requiring point values and derivatives to be interpolated on the face-neighboring cells. The resulting over-determined linear system of equations is then solved in the least-squares sense. This reconstruction scheme only involves the von Neumann neighborhood, and thus is compact, simple, robust, and flexible. Furthermore, the reconstruction scheme guarantees exact conservation, not only of the cell averages but also of their slopes due to a judicious choice of our Taylor basis.

More recently, Zhang et al.^{38,39} presented a class of hybrid DG/FV methods for the conservation laws, where the second derivatives in a cell are obtained from the first derivatives in the cell itself and its neighboring cells using a Green-Gauss reconstruction widely used in the finite volume methods. This provides a fast, simple, and robust way to obtain a higher-order polynomial solutions. The numerical experiments indicate that this efficient reconstruction scheme is able to achieve a third-order accuracy: one order accuracy higher than the underlying second order DG method.

The objective of the effort discussed in this paper is to conduct a comparative study on these three reconstruction/recovery methods for computing compressible flows on arbitrary grids. A variety of flow problems are performed in order to access the accuracy of these three reconstruction methods. The numerical experiments demonstrate that all three reconstruction methods can significantly improve the accuracy of the underlying second-order DG method, although the least-squares reconstruction method provides the best performance in terms of both accuracy and robustness. The remainder of this paper is organized as follows. The governing equations are described in Section 2. The underlying reconstructed discontinuous Galerkin methods are presented in Section 3. Extensive numerical experiments are reported in Section 4. Concluding remarks are given in Section 5.

II. Governing Equations

The Navier-Stokes equations governing unsteady compressible viscous flows can be expressed as

$$\frac{\partial \mathbf{U}(x, t)}{\partial t} + \frac{\partial \mathbf{F}_k(\mathbf{U}(x, t))}{\partial x_k} = \frac{\partial \mathbf{G}_k(\mathbf{U}(x, t))}{\partial x_k} \quad (2.1)$$

where the summation convention has been used. The conservative variable vector \mathbf{U} , advective (inviscid) flux vector \mathbf{F} , and viscous flux vector \mathbf{G} are defined by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_i \\ \rho e \end{pmatrix} \quad \mathbf{F}_j = \begin{pmatrix} \rho u_j \\ \rho u_i u_j + p \delta_{ij} \\ u_j (\rho e + p) \end{pmatrix} \quad \mathbf{G}_j = \begin{pmatrix} 0 \\ \sigma_{ij} \\ u_i \sigma_{lj} + q_j \end{pmatrix} \quad (2.2)$$

Here ρ , p , and e denote the density, pressure, and specific total energy of the fluid, respectively, and u_i is the velocity of the flow in the coordinate direction x_i . The pressure can be computed from the equation of state

$$p = (\gamma - 1) \rho \left(e - \frac{1}{2} u_j u_j \right) \quad (2.3)$$

which is valid for perfect gas, where γ is the ratio of the specific heats. The components of the viscous stress tensor σ_{ij} and the heat flux vector are given by

$$\sigma_{ij} = \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) - \frac{2}{3} \mu \frac{\partial u_k}{\partial x_k} \delta_{ij} \quad q_j = \frac{1}{\gamma - 1} \frac{\mu}{\text{Pr}} \frac{\partial T}{\partial x_j} \quad (2.4)$$

In the above equations, T is the temperature of the fluid, Pr the laminar Prandtl number, which is taken as 0.7 for air. μ represents the molecular viscosity, which is determined through Sutherland's law

$$\frac{\mu}{\mu_0} = \left(\frac{T}{T_0} \right)^{\frac{3}{2}} \frac{T_0 + S}{T + S} \quad (2.5)$$

μ_0 denotes the viscosity at the reference temperature T_0 , and S is a constant which for air assumes the value $S = 110^\circ\text{K}$. The temperature of the fluid T is determined by

$$T = \gamma \frac{p}{\rho} \quad (2.6)$$

Neglecting viscous effects, the left-hand side of Eq. (2.1) represents the Euler equations governing unsteady compressible inviscid flows.

III. Reconstructed Discontinuous Galerkin Methods

The governing equation (2.1) is discretized using a discontinuous Galerkin finite element formulation. To formulate the discontinuous Galerkin method, we first introduce the following weak formulation, which is obtained by multiplying the above conservation law by a test function \mathbf{W} , integrating over the domain Ω , and then performing an integration by parts,

$$\int_{\Omega} \frac{\partial \mathbf{U}}{\partial t} \mathbf{W} d\Omega + \int_{\Gamma} \mathbf{F}_k \mathbf{n}_k d\Gamma - \int_{\Omega} \mathbf{F}_k \frac{\partial \mathbf{W}}{\partial x_k} d\Omega = \int_{\Gamma} \mathbf{G}_k \mathbf{n}_k d\Gamma - \int_{\Omega} \mathbf{G}_k \frac{\partial \mathbf{W}}{\partial x_k} d\Omega, \quad \forall \mathbf{W} \in V \quad (3.1)$$

where $\Gamma(=\partial\Omega)$ denotes the boundary of Ω , and \mathbf{n}_j the unit outward normal vector to the boundary. We assume that the domain Ω is subdivided into a collection of non-overlapping elements Ω_e , which can be triangles, quadrilaterals, polygons, or their combinations in 2D and tetrahedra, prisms, pyramids, and hexahedra or their combinations in 3D. We introduce the following broken Sobolev space V_h^p

$$V_h^p = \left\{ v_h \in [L_2(\Omega)]^m : v_h|_{\Omega_e} \in [V_p^m] \forall \Omega_e \in \Omega \right\} \quad (3.2)$$

which consists of discontinuous vector-values polynomial functions of degree p , and where m is the dimension of the unknown vector and

$$V_p^m = \text{span} \left\{ \prod x_i^{\alpha_i} : 0 \leq \alpha_i \leq p, 0 \leq i \leq d \right\} \quad (3.3)$$

where α denotes a multi-index and d is the dimension of space. Then, we can obtain the following semi-discrete form by applying weak formulation on each element Ω_e

$$\begin{aligned} & \text{Find } \mathbf{U}_h \in V_h^p \quad \text{such as} \\ & \frac{d}{dt} \int_{\Omega_e} \mathbf{U}_h \mathbf{W}_h d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k \mathbf{W}_h d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}_h) \frac{\partial \mathbf{W}_h}{\partial x_k} d\Omega = \\ & \int_{\Gamma_e} \mathbf{G}_k(\mathbf{U}_h) \mathbf{n}_k \mathbf{W}_h d\Gamma - \int_{\Omega_e} \mathbf{G}_k(\mathbf{U}_h) \frac{\partial \mathbf{W}_h}{\partial x_k} d\Omega, \quad \forall \mathbf{W}_h \in V_h^p \end{aligned} \quad (3.4)$$

where \mathbf{U}_h and \mathbf{W}_h represent the finite element approximations to the analytical solution \mathbf{U} and the test function \mathbf{W} respectively, and they are approximated by a piecewise polynomial function of degrees p , which are discontinuous between the cell interfaces. Assume that B is the basis of polynomial function of degrees p , this is then equivalent to the following system of N equations,

$$\begin{aligned} & \frac{d}{dt} \int_{\Omega_e} \mathbf{U}_h B_i d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k B_i d\Gamma - \int_{\Omega_e} \mathbf{F}_k(\mathbf{U}_h) \frac{\partial B_i}{\partial x_k} d\Omega = \\ & \int_{\Gamma_e} \mathbf{G}_k(\mathbf{U}_h) \mathbf{n}_k B_i d\Gamma - \int_{\Omega_e} \mathbf{G}_k(\mathbf{U}_h) \frac{\partial B_i}{\partial x_k} d\Omega, \quad 1 \leq i \leq N \end{aligned} \quad (3.5)$$

where N is the dimension of the polynomial space. Since the numerical solution \mathbf{U}_h is discontinuous between element interfaces, the interface fluxes are not uniquely defined. The flux function $\mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k$ appearing in the second terms of Eq. (3.5) is replaced by a numerical Riemann flux function $\mathbf{H}_k(\mathbf{U}_h^L, \mathbf{U}_h^R, \mathbf{n}_k)$ where \mathbf{U}_h^L and \mathbf{U}_h^R are the conservative state vector at the left and right side of the element boundary. The computation of the viscous fluxes in the boundary integral has to properly resolve the discontinuities at the interfaces. This scheme is called discontinuous Galerkin method of degree p , or in short notation DG(P) method. Note that discontinuous Galerkin formulations are very similar to finite volume schemes, especially in their use of numerical fluxes. Indeed, the classical first-order cell-centered finite volume scheme exactly corresponds to the DG(P₀) method, i.e., to the discontinuous Galerkin method using a piecewise constant polynomial. Consequently, the DG(P_k) methods with $k > 0$ can be regarded as a natural generalization of finite volume methods to higher order methods. By simply increasing the degree P of the polynomials, the DG methods of corresponding higher order are obtained.

The domain and boundary integrals in Eq. (3.5) are calculated using Gauss quadrature formulas. The number of quadrature points used is chosen to integrate exactly polynomials of order of $2p$ and $2p+1$ on the reference element for the domain and boundary integrals, respectively. In 2D, two, three, and four points are used for linear, quadratic, and cubic basis function in the boundary integrals. The domain integrals are evaluated using three, six, and thirteen points for triangular elements and four, nine, and sixteen points for quadrilateral elements, respectively.

In the traditional DGM, numerical polynomial solutions \mathbf{U}_h in each element are expressed using either standard Lagrange finite element or hierarchical node-based basis as following

$$\mathbf{U}_h = \sum_{i=1}^N \mathbf{U}_i(t) B_i(x), \quad (3.6)$$

where B_i are the finite element basis functions. As a result, the unknowns to be solved are the variables at the nodes U_i , as illustrated in Figure 1 for linear and quadratic polynomial approximations.

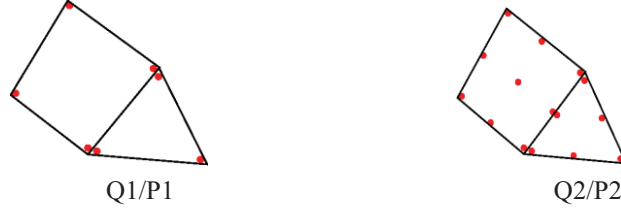


Figure. 1. Representation of polynomial solutions using finite element shape functions

On each cell, a system of $N \times N$ has to be solved, where polynomial solutions are dependent on the shape of elements. For example, for a linear polynomial approximation in 2D as shown in Fig.1, a linear polynomial is used for triangular elements and the unknowns to be solved are the variables at the three vertices and a bi-linear polynomial is used for quadrilateral elements and the unknowns to be solved are the variables at the four vertices. However, numerical polynomial solutions U can be expressed in other forms as well. In the present work, the numerical polynomial solutions are represented using a Taylor series expansion at the center of the cell. For example, if we do a Taylor series expansion at the cell centroid, the quadratic polynomial solutions can be expressed as follows

$$U_h = U_c + \frac{\partial U}{\partial x} \Big|_c (x - x_c) + \frac{\partial U}{\partial y} \Big|_c (y - y_c) + \frac{\partial^2 U}{\partial x^2} \Big|_c \frac{(x - x_c)^2}{2} + \frac{\partial^2 U}{\partial y^2} \Big|_c \frac{(y - y_c)^2}{2} + \frac{\partial^2 U}{\partial x \partial y} \Big|_c (x - x_c)(y - y_c) \quad (3.7)$$

which can be further expressed as cell-averaged values and their derivatives at the center of the cell:

$$U_h = \tilde{U} + \frac{\partial U}{\partial x} \Big|_c (x - x_c) + \frac{\partial U}{\partial y} \Big|_c (y - y_c) + \frac{\partial^2 U}{\partial x^2} \Big|_c \left(\frac{(x - x_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\Omega \right) + \frac{\partial^2 U}{\partial y^2} \Big|_c \left(\frac{(y - y_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\Omega \right) + \frac{\partial^2 U}{\partial x \partial y} \Big|_c \left((x - x_c)(y - y_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\Omega \right) \quad (3.8)$$

where \tilde{U} is the mean value of U in this cell. The unknowns to be solved in this formulation are the cell-averaged variables and their derivatives at the center of the cells, regardless of element shapes, as shown in Fig.2.

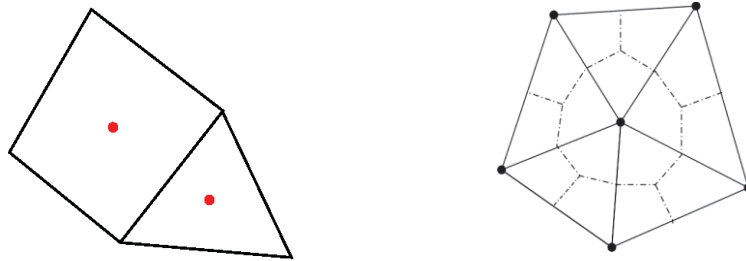


Figure 2. Representation of polynomial solutions using a Taylor series expansion for a cell-centered scheme (left) and vertex-centered scheme (right)

In this case, the dimension of the polynomial space is six and the six basis functions are

$$\begin{aligned} B_1 &= 1 & B_2 &= x - x_c & B_3 &= y - y_c & B_4 &= \frac{(x - x_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2} d\Omega \\ B_5 &= \frac{(y - y_c)^2}{2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2} d\Omega & B_6 &= (x - x_c)(y - y_c) - \frac{1}{\Omega_e} \int_{\Omega_e} (x - x_c)(y - y_c) d\Omega \end{aligned} \quad (3.9)$$

The discontinuous Galerkin formulation then leads to the following six equations

$$\begin{aligned} \frac{d}{dt} \int_{\Omega_e} \tilde{\mathbf{U}} d\Omega + \int_{\Gamma_e} \mathbf{F}_k(\mathbf{U}_h) \mathbf{n}_k d\Gamma &= 0, \quad i = 1 \\ M_{5 \times 5} \frac{d}{dt} \left(\frac{\partial \mathbf{U}}{\partial x} \Big|_c, \frac{\partial \mathbf{U}}{\partial y} \Big|_c, \frac{\partial^2 \mathbf{U}}{\partial x^2} \Big|_c, \frac{\partial^2 \mathbf{U}}{\partial y^2} \Big|_c, \frac{\partial^2 \mathbf{U}}{\partial x \partial y} \Big|_c \right)^T + R_{5 \times 1} &= 0 \end{aligned} \quad (3.10)$$

Note that in this formulation, equations for the cell-averaged variables are decoupled from equations for their derivatives due to the judicious choice of the basis functions and the fact that

$$\int_{\Omega_e} B_i B_j d\Omega = 0, \quad 2 \leq i \leq 6 \quad (3.11)$$

In the implementation of this DG method, the basis functions are actually normalized in order to improve the conditioning of the system matrix (3.5) as follows:

$$\begin{aligned} B_1 &= 1 & B_2 &= \frac{x - x_c}{\Delta x} & B_3 &= \frac{y - y_c}{\Delta y} & B_4 &= \frac{(x - x_c)^2}{2\Delta x^2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)^2}{2\Delta x^2} d\Omega \\ B_5 &= \frac{(y - y_c)^2}{2\Delta y^2} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(y - y_c)^2}{2\Delta y^2} d\Omega & B_6 &= \frac{(x - x_c)(y - y_c)}{\Delta x \Delta y} - \frac{1}{\Omega_e} \int_{\Omega_e} \frac{(x - x_c)(y - y_c)}{\Delta x \Delta y} d\Omega \end{aligned} \quad (3.12)$$

where $\Delta x = 0.5(x_{\max} - x_{\min})$, and $\Delta y = 0.5(y_{\max} - y_{\min})$, and x_{\max} , x_{\min} , y_{\max} , and y_{\min} are the maximum and minimum coordinates in the cell Ω_e in x-, and y-directions, respectively. A quadratic polynomial solution can then be rewritten as

$$\mathbf{U}_h = \tilde{\mathbf{U}} + \frac{\partial \mathbf{U}}{\partial x} \Big|_c \Delta x B_2 + \frac{\partial \mathbf{U}}{\partial y} \Big|_c \Delta y B_3 + \frac{\partial^2 \mathbf{U}}{\partial x^2} \Big|_c \Delta x^2 B_4 + \frac{\partial^2 \mathbf{U}}{\partial y^2} \Big|_c \Delta y^2 B_5 + \frac{\partial^2 \mathbf{U}}{\partial x \partial y} \Big|_c \Delta x \Delta y B_6 \quad (3.13)$$

The above normalization is especially important to alleviate the stiffness of the system matrix for higher-order DG approximations.

This formulation allows us to clearly see the similarities and differences between the DG and FV methods. In fact, the discretized governing equations for the cell-averaged variables and the assumption of polynomial solutions on each cell are exactly the same for both methods. The only difference between them is the way how they obtain high-order (>1) polynomial solutions. In the finite volume methods, the polynomial solution of degrees p are reconstructed using information from the cell-averaged values of the flow variables, which can be obtained using either TVD/MUSCL or ENO/WENO reconstruction schemes. Unfortunately, the multi-dimensional MUSCL approach suffers from two shortcomings in the context of unstructured grids: 1) Uncertainty and arbitrariness in choosing the stencils and methods to compute the gradients in the case of linear reconstruction; This explains why a nominally second-order finite volume scheme is hardly able to deliver a formal solution of the second order accuracy in practice for unstructured grids. The situation becomes even more evident, severe, and profound, when a highly stretched tetrahedral grid is used in the boundary layers. Many studies, as reported by many researchers²⁶⁻²⁸ have demonstrated that it is difficult to obtain a second-order accurate flux reconstruction on highly stretched tetrahedral grids and that for the discretization of inviscid fluxes, the classic 1D-based upwind schemes using median-dual finite volume approximation suffer from excessive numerical diffusion due to such skewing. 2) Extended stencils required for the reconstruction of higher-order ($>1^{\text{st}}$) polynomial solutions. This is exactly the reason why the current finite-volume methods using the TVD/MUSCL reconstruction are not practical at higher order and have remained second-order on unstructured grids. When the ENO/WENO reconstruction schemes are used for the construction of a polynomial of degree p on unstructured grids, the dimension of the polynomial space $N = N(p, d)$ depends on the degree of the polynomials of the expansion p , and the number of spatial dimensions d . One must have three, six, and ten cells in 2D and four, ten, and twenty cells in 3D for the construction of a linear, quadratic, cubic Lagrange polynomial, respectively. Undoubtedly, it is an overwhelmingly challenging, if not practically impossible, task to judiciously choose a set of admissible and proper stencils that have such a large number of cells on unstructured grids especially for higher order polynomials and higher dimensions. This explains why the application of higher-order ENO/WENO methods hardly exists on unstructured grids, in spite of their

tremendous success on structured grids and their superior performance over the MUSCL/TVD methods. Unlike the FV methods, where the derivatives are reconstructed using cell average values of the neighboring cells, the DG method computes the derivatives in a manner similar to the mean variables. This is compact, rigorous, and elegant mathematically in contrast with arbitrariness characterizing the reconstruction schemes with respect how to compute the derivatives and how to choose the stencils used in the FV methods. It is our believe that this is one of the main reasons why the second order DG methods are more accurate than the FV methods using either TVD/MUSCL or ENO/WENO reconstruction schemes and are less dependent on the mesh regularity, which has been demonstrated numerically¹³. Furthermore, the higher order DG methods can be easily constructed by simply increasing the degree p of the polynomials locally, in contrast to the finite volume methods which use the extended stencils to achieve higher order of accuracy.

However, in comparison with reconstructed FV methods, the DG methods have a significant drawback in that they require more degrees of freedom, an additional domain integration, and more Gauss quadrature points for the boundary integration, and therefore more computational costs and storage requirements. On one hand, the reconstruction methods that FV methods use to achieve higher-order accuracy are relatively inexpensive but less accurate and robust. On the other hand, DG methods that can be viewed as a different way to extend a FV method to higher orders are accurate and robust but costly. It is only natural and tempting to combine the efficiency of the reconstruction methods and the accuracy of the DG methods. This idea was originally introduced by Dumbser et al in the frame of PnPm scheme¹⁸⁻²⁰, where Pn indicates that a piecewise polynomial of degree of n is used to represent a DG solution, and Pm represents a reconstructed polynomial solution of degree of m ($m \geq n$) that is used to compute the fluxes and source terms. The beauty of PnPm schemes is that they provide a unified formulation for both finite volume and DG methods, and contain both classical finite volume and standard DG methods as two special cases of PnPm schemes, and thus allow for a direct efficiency comparison. When $n=0$, i.e. a piecewise constant polynomial is used to represent a numerical solution, P0Pm is nothing but classical high order finite volume schemes, where a polynomial solution of degree m ($m \geq 1$) is reconstructed from a piecewise constant solution. When $m=n$, the reconstruction reduces to the identity operator, and PnPm scheme yields a standard DG method. Clearly, an accurate and efficient reconstruction is the key ingredient in extending the underlying DG method to higher order accuracy. Although our discussion in this work is mainly focused on the linear DG method, its extension to higher order DG methods is straightforward. In the case of DG(P1) method, a linear polynomial solution U_i in any cell i is

$$U_i = \tilde{U}_i + \frac{\partial U}{\partial x} \Big|_i \Delta x_i B_2 + \frac{\partial U}{\partial y} \Big|_i \Delta y_i B_3 \quad (3.14)$$

Using this underlying linear polynomial DG solution in the neighboring cells, one can reconstruct a quadratic polynomial solution U_i^R as follows:

$$U_i^R = \tilde{U}_i^R + \frac{\partial U^R}{\partial x} \Big|_i \Delta x_i B_2 + \frac{\partial U^R}{\partial y} \Big|_i \Delta y_i B_3 + \frac{\partial^2 U^R}{\partial x^2} \Big|_i \Delta x_i^2 B_4 + \frac{\partial^2 U^R}{\partial y^2} \Big|_i \Delta y_i^2 B_5 + \frac{\partial^2 U^R}{\partial x \partial y} \Big|_i \Delta x_i \Delta y_i B_6 \quad (3.15)$$

In order to maintain the compactness of the DG methods, the reconstruction is required to only involve Von Neumann neighborhood, i.e., the adjacent cells that share a face with the cell i under consideration. There are six degrees of freedom, and therefore 6 unknowns to be determined. However, the first three unknowns can be trivially obtained, by requiring that the reconstruction scheme has to be conservative, a fundamental requirement, and the values of the reconstructed first derivatives are equal to the ones of the first derivatives of the underlying DG solution at the centroid i . Due to the judicious choice of Taylor basis in our DG formulation, these three degrees of freedom (cell average and slopes) simply coincide with the ones from the underlying DG solution, i.e.,

$$\tilde{U}_i^R = \tilde{U}_i \quad \frac{\partial U^R}{\partial x} \Big|_i = \frac{\partial U}{\partial x} \Big|_i \quad \frac{\partial U^R}{\partial y} \Big|_i = \frac{\partial U}{\partial y} \Big|_i \quad (3.16)$$

Three methods of determining the remaining three degrees of freedom: the second derivatives (curvatures) will be addressed and discussed in this section.

A. Least-Squares Recovery

The least-square recovery scheme by Dumbser et al¹⁸⁻²⁰ relies on L_2 -projection to determine the second derivatives. Consider a face-neighboring cell j , the recovery principle requires

$$\int_{\Omega_j} U_i^R B_k^j d\Omega = \int_{\Omega_j} U_j B_k^j d\Omega, \quad 1 \leq k \leq 3 \quad (3.17)$$

where B_k^j ($k=1,2,3$) are the three basis function on cell j . Note that during the recovery process, the recovered polynomial solution is continuously extended over the face-neighboring cells, and the locally recovered solution is indistinguishable from the underlying DG solution in the neighboring cells in the weak sense. The same recovery equations constituted by Eq. (3.17) can be written for all cells connected to the cell i with a common face, which leads to a non-square matrix. The number of face-neighboring cells for a triangular and quadrilateral cell is three and four, respectively. As a result, the size of the resulting non-square matrix is 9×3 and 12×3 , respectively. This over-determined linear system of 9 or 12 equations for 3 unknowns can be solved in the least-squares sense. One disadvantage of the recovery is the need to compute the integral on the left-hand-side of the recovered equations, which is done using the classical multidimensional Gaussian quadrature of an appropriate order. Furthermore, the recovery might be problematic for a boundary cell, where the number of the face-neighboring cells might be not enough to recover a polynomial solution of a desired order. For example, a corner tetrahedral cell with three boundary faces has only one face-neighboring cell, which can only provide four recovered equations. However, there exist six second derivatives in 3D. In this case, one cannot recover a quadratic polynomial solution from the underlying linear discontinuous Galerkin solution. This situation can be readily remedied by use of extended one-sided stencils, although the compactness of the underlying DG methods is then sacrificed. The resulting reconstructed DG method will be termed as P1P2(rc) method in this paper.

B. Least-Squares Reconstruction

Alternatively, the remaining three degrees of freedom can be determined by requiring that the reconstructed solution and its first derivatives are equal to the underlying DG solution and its first derivatives for all the adjacent face neighboring cells. Consider a neighboring cell j , one requires

$$\begin{aligned} \mathbf{U}_j &= \tilde{\mathbf{U}}_i^R + \frac{\partial \mathbf{U}^R}{\partial x} \Big|_i \Delta x_i B_2 + \frac{\partial \mathbf{U}^R}{\partial y} \Big|_i \Delta y_i B_3 + \frac{\partial^2 \mathbf{U}^R}{\partial x^2} \Big|_i \Delta x_i^2 B_4 + \frac{\partial^2 \mathbf{U}^R}{\partial y^2} \Big|_i \Delta y_i^2 B_5 + \frac{\partial^2 \mathbf{U}^R}{\partial x \partial y} \Big|_i \Delta x_i \Delta y_i B_6 \\ \frac{\partial \mathbf{U}}{\partial x} \Big|_j &= \frac{\partial \mathbf{U}^R}{\partial x} \Big|_i \Delta x_i \frac{1}{\Delta x_i} + \frac{\partial^2 \mathbf{U}^R}{\partial x^2} \Big|_i \Delta x_i^2 \frac{B_2}{\Delta x_i} + \frac{\partial^2 \mathbf{U}^R}{\partial x \partial y} \Big|_i \Delta x_i \Delta y_i \frac{B_3}{\Delta x_i} \\ \frac{\partial \mathbf{U}}{\partial y} \Big|_j &= \frac{\partial \mathbf{U}^R}{\partial y} \Big|_i \Delta y_i \frac{1}{\Delta y_i} + \frac{\partial^2 \mathbf{U}^R}{\partial y^2} \Big|_i \Delta y_i^2 \frac{B_3}{\Delta y_i} + \frac{\partial^2 \mathbf{U}^R}{\partial x \partial y} \Big|_i \Delta x_i \Delta y_i \frac{B_2}{\Delta y_i} \end{aligned} \quad (3.18)$$

where the basis function B are evaluated at the center of cell j , i.e., $B = \mathbf{B}(x_j, y_j)$. This can be written in a matrix form as follows:

$$\begin{pmatrix} B_4^j & B_5^j & B_6^j \\ B_2^j & 0 & B_3^j \\ 0 & B_3^j & B_2^j \end{pmatrix} \begin{pmatrix} \frac{\partial^2 \mathbf{U}^R}{\partial x^2} \Big|_i \Delta x_i^2 \\ \frac{\partial^2 \mathbf{U}^R}{\partial y^2} \Big|_i \Delta y_i^2 \\ \frac{\partial^2 \mathbf{U}^R}{\partial x \partial y} \Big|_i \Delta x_i \Delta y_i \end{pmatrix} = \begin{pmatrix} \mathbf{U}_j - (\mathbf{U}_i B_1^j + \frac{\partial \mathbf{U}^R}{\partial x} \Big|_i \Delta x_i B_2^j + \frac{\partial \mathbf{U}^R}{\partial y} \Big|_i \Delta y_i B_3^j) \\ \frac{\Delta x_i}{\Delta x_j} \frac{\partial \mathbf{U}^R}{\partial x} \Big|_i \Delta x_j - \frac{\partial \mathbf{U}^R}{\partial x} \Big|_i \Delta x_i \\ \frac{\Delta y_i}{\Delta y_j} \frac{\partial \mathbf{U}^R}{\partial y} \Big|_i \Delta y_j - \frac{\partial \mathbf{U}^R}{\partial y} \Big|_i \Delta y_i \end{pmatrix} = \begin{pmatrix} \mathbf{R}_1^j \\ \mathbf{R}_2^j \\ \mathbf{R}_3^j \end{pmatrix} \quad (3.19)$$

where \mathbf{R} is used to represent the right-hand-side for simplicity. Similar equations could be written for all cells connected to the cell i with a common face, which leads to a non-square matrix. The number of face-neighboring cells for a triangular and quadrilateral cell is three and four, respectively. As a result, the size of the resulting non-square matrix is 9×3 and 12×3 , respectively. This over-determined linear system of 9 or 12 equations for 3 unknowns can be solved in the least-squares sense. In the present work, it is solved using a normal equation approach, which, by pre-multiplying through by matrix transpose, yields a symmetric linear system of equations 3×3 as follows

$$\begin{pmatrix} \sum_j (B_4^j B_4^j + B_4^j B_4^j) & \sum_j B_4^j B_5^j & \sum_j (B_4^j B_6^j + B_2^j B_3^j) \\ \sum_j B_4^j B_5^j & \sum_j (B_5^j B_5^j + B_3^j B_3^j) & \sum_j (B_5^j B_6^j + B_2^j B_3^j) \\ \sum_j (B_4^j B_6^j + B_2^j B_3^j) & \sum_j (B_5^j B_6^j + B_2^j B_3^j) & \sum_j (B_6^j B_6^j + B_2^j B_2^j + B_3^j B_3^j) \end{pmatrix} \begin{pmatrix} \frac{\partial^2 \mathbf{U}^R}{\partial x^2} \Big|_i \Delta x_i^2 \\ \frac{\partial^2 \mathbf{U}^R}{\partial y^2} \Big|_i \Delta y_i^2 \\ \frac{\partial^2 \mathbf{U}^R}{\partial x \partial y} \Big|_i \Delta x_i \Delta y_i \end{pmatrix} \\
= \begin{pmatrix} \sum_j (B_4^j \mathbf{R}_1^j + B_2^j \mathbf{R}_2^j) \\ \sum_j (B_5^j \mathbf{R}_1^j + B_3^j \mathbf{R}_3^j) \\ \sum_j (B_6^j \mathbf{R}_1^j + B_3^j \mathbf{R}_2^j + B_2^j \mathbf{R}_3^j) \end{pmatrix} \quad (3.20)$$

This linear system of 3x3 can be then trivially solved to obtain the second derivatives of the reconstructed quadratic polynomial solution. This reconstructed DG method will be referred as P1P2(RC) method from now on.

C. Green-Gauss Reconstruction

Green-Gauss reconstruction is probably the simplest reconstruction scheme, which is mostly used to reconstruct a gradient from the cell-averaged values in the finite volume methods. Similarly, the second derivatives in a cell i can be reconstructed from the known first derivatives using Green's theorem as follows,

$$\int_{\Omega_i} \frac{\partial^2 \mathbf{U}}{\partial x^2} d\Omega = \Omega_i \frac{\partial^2 \mathbf{U}}{\partial x^2} \Big|_i = \int_{\Gamma_i} \frac{\partial \mathbf{U}}{\partial x} n_x d\Gamma \quad (3.21)$$

$$\int_{\Omega_i} \frac{\partial^2 \mathbf{U}}{\partial y^2} d\Omega = \Omega_i \frac{\partial^2 \mathbf{U}}{\partial y^2} \Big|_i = \int_{\Gamma_i} \frac{\partial \mathbf{U}}{\partial y} n_y d\Gamma \quad (3.22)$$

$$\int_{\Omega_i} \frac{\partial^2 \mathbf{U}}{\partial x \partial y} d\Omega = \Omega_i \frac{\partial^2 \mathbf{U}}{\partial x \partial y} \Big|_i = \int_{\Gamma_i} \frac{\partial \mathbf{U}}{\partial x} n_y d\Gamma \quad (3.23)$$

$$\int_{\Omega_i} \frac{\partial^2 \mathbf{U}}{\partial y \partial x} d\Omega = \Omega_i \frac{\partial^2 \mathbf{U}}{\partial y \partial x} \Big|_i = \int_{\Gamma_i} \frac{\partial \mathbf{U}}{\partial y} n_x d\Gamma \quad (3.24)$$

The cross-derivatives can be computed using either equation 3.23 or equation 3.24. In the present work, an arithmetic mean is used to obtain the cross-derivatives. In Eqs. 3.22, 3.23, 3.24, and 3.25, the first derivatives at the interface are computed using a simple volume-weighted average. The beauty of this reconstruction scheme is its simplicity, efficiency, and robustness, since it does not need to solve an over-determined system using a least-squares approach. However, the Green-Gauss reconstruction is not as accurate as its least-squares recovery/reconstruction counterparts, as it only involves the first derivatives of the underlying DG solutions. This reconstructed DG method is named P1P2(GG) hereafter.

In the reconstructed DG methods, the reconstructed quadratic polynomial solution is then used to compute the domain and boundary integrals of the underlying DG(P1) method in Eq. (3.5). The resulting DG method, termed a reconstructed DG method (RDG(P1P2) or P1P2 in short notation), is expected to have a third order of accuracy at a moderate increase of computing costs in comparison with the underlying DG(P1) method. The extra costs are mainly due to the reconstruction step, which is relatively cheap in comparison to the evaluation of fluxes, and an extra Gauss quadrature point, which is required to calculate the domain integrals for the triangular element (four quadrature points). Like DG(P1) methods, two quadrature points are used to calculate the boundary integrals, and four points are used to calculate the domain integrals for quadrilateral elements. In comparison to DG(P2) methods,

this represents a significant savings in terms of flux evaluations. Furthermore, the number of degrees of freedom is significantly reduced, which leads to a significant reduction in memory requirements, and from which implicit methods will benefit tremendously.

IV. Numerical Examples

A. Convection of an isentropic vortex

The convection of a 2D inviscid isentropic given for example by Dumbser et al¹⁸ is considered in this test case to conduct a convergence study of the RDG methods. The analytical solution to this problem at any time t is simply the passive advection of the initial solution at $t=0$, which provides a valuable reference for measuring the accuracy of a numerical solution. The initial condition is a linear superposition of a mean uniform flow with some perturbations δ . The free stream flow conditions are $(\rho_\infty, u_\infty, v_\infty, p_\infty) = (1, 1, 1, 1)$. The perturbations of the velocity components u and v , entropy S , and temperature T for the vortex are given by

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-y_0) \\ (x-x_0) \end{pmatrix}, \quad \delta S = 0, \quad \delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2},$$

where $r^2 = (x-x_0)^2 + (y-y_0)^2$, (x_0, y_0) is the coordinate of the vortex center, and ε is the vortex strength. From $\rho = \rho_\infty + \delta\rho$, $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T = T_\infty + \delta T$, and the isentropic relation, other physical variables can be determined as follows

$$\rho = T^{\frac{1}{\gamma-1}} = (T_\infty + \delta T)^{\frac{1}{\gamma-1}} = \left[T_\infty - \frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2} e^{1-r^2} \right]^{\frac{1}{\gamma-1}},$$

$$\rho u = \rho(u_\infty + \delta u) = \rho \left[u_\infty - \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (y-y_0) \right],$$

$$\rho v = \rho(v_\infty + \delta v) = \rho \left[v_\infty + \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} (x-x_0) \right],$$

$$p = \rho^\gamma,$$

$$e = \frac{p}{\rho(\gamma-1)} + \frac{1}{2}(u^2 + v^2).$$

In this test case, the vortex strength $\varepsilon=5$, and the coordinate of the vortex center (x_0, y_0) is $(5, 5)$. The computational domain Ω is $[0, 10] \times [0, 10]$ and the periodic boundary conditions are imposed. The numerical solutions are obtained after one period of time $t=10$, and compared with the exact solution simply given by the initial condition. The follow L_2 -norm

$$\|\rho^R - \rho^e\|_{L^2(\Omega)} = \left(\int_{\Omega} |\rho_p^R - \rho^e|^2 d\Omega \right)^{\frac{1}{2}},$$

is used to measure the error between the numerical and analytical solutions, where ρ^R is the reconstructed quadratic solution for the density. Figure 3 shows three successively refined triangular grids having 554, 2,216, and 8,864 triangles, respectively. The number of faces in each space dimension is 16, 32, and 64, respectively. Figure 4 provides the details of the spatial convergence of the three RDG methods for this numerical experiment. As expected, all three reconstructed (P1P2) methods offer a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P1) method.

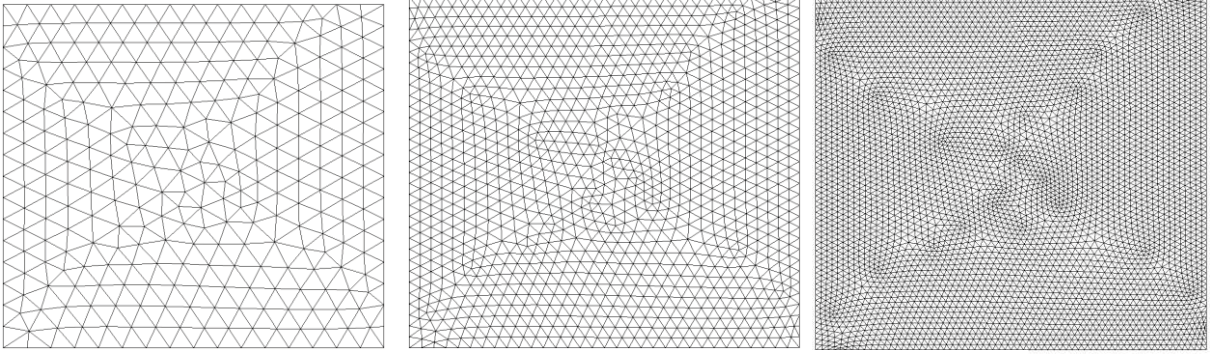


Figure 3. Sequence of triangular grids used for computing the convection of an isentropic vortex.

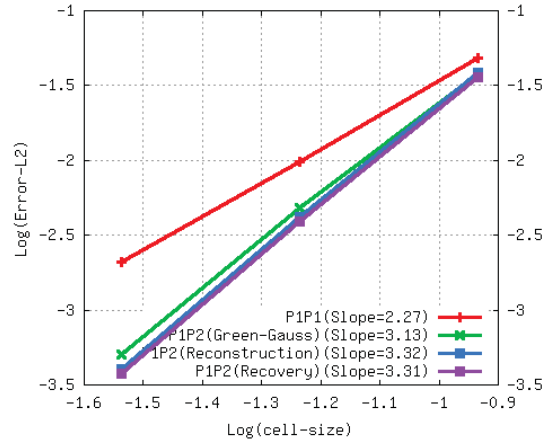


Figure 4. Comparison of the convergence histories for different RDG method for the convection of an isentropic vortex.

B. Subsonic Flows past a Circular Cylinder

This is a well-known test case: subsonic flow past a circular cylinder at a Mach number of $M_\infty=0.38$. A set of three hybrid grids is used in this test case to verify if a formal order of the convergence rate of the three RDG methods can be achieved for the compressible Euler equations on hybrid grids. Figure 5 shows three successively refined hybrid grids having 32×9 , 64×17 , and 128×33 points, respectively. The first number is the number of points in the angular direction, and the second number is the number of points in the radial direction. The radius of the cylinder is $r_1=0.5$, the domain is bounded by $r_{33}=20$, and the radii of concentric circles for 128×33 mesh are set up as

$$r_i = r_1 \left(1 + \frac{2\pi}{128} \sum_{j=0}^{i-1} \alpha^j \right), \quad i = 2, \dots, 33$$

where $\alpha=1.1580372$. The coarser grids are generated by successively coarsening the finest mesh. Numerical solutions to this problem are computed using the three reconstruction RDG(P1P2) methods on these three grids to obtain quantitative measurement of the order of accuracy and discretization errors. In this case, the L_2 -norm of the following entropy production ε defined as

$$\varepsilon = \frac{S - S_\infty}{S_\infty} = \frac{p}{p_\infty} \left(\frac{\rho_\infty}{\rho} \right)^\gamma - 1$$

is served as the error measurement, where S is the entropy. Note that the entropy production is a very good criterion to measure accuracy of the numerical solutions, since the flow under consideration is isentropic. Figures 6 and 7

show the computed density contours in the flow field using these three grids obtained by DG(P1) and P1P2(RC) method, respectively, where one can clearly observe the significant improvement of the computed solution using the RDG method. Figure 8 provides the details of the spatial convergence of each reconstruction method for this numerical experiment. As expected, all reconstructed RDG(P1P2) methods offer a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P1) method. However, the least-squares reconstruction and recovery methods are better than the Green-Gauss reconstruction method in terms of both absolute error and order of convergence, and the P1P2(RC) method outperforms P1P2(rc) method in this case.

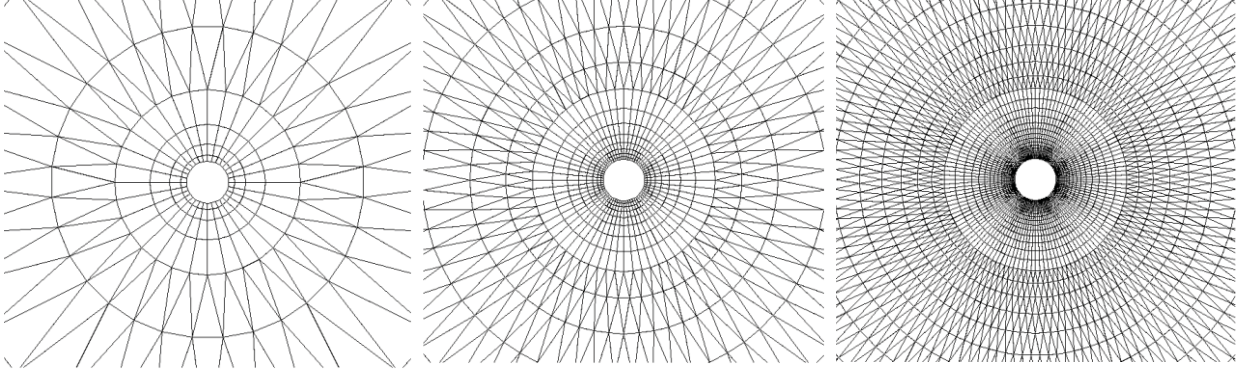


Figure 5: Sequences of three successively globally refined hybrid meshes 32x9, 64x17, 128x33 for computing subsonic flow past a circular cylinder.

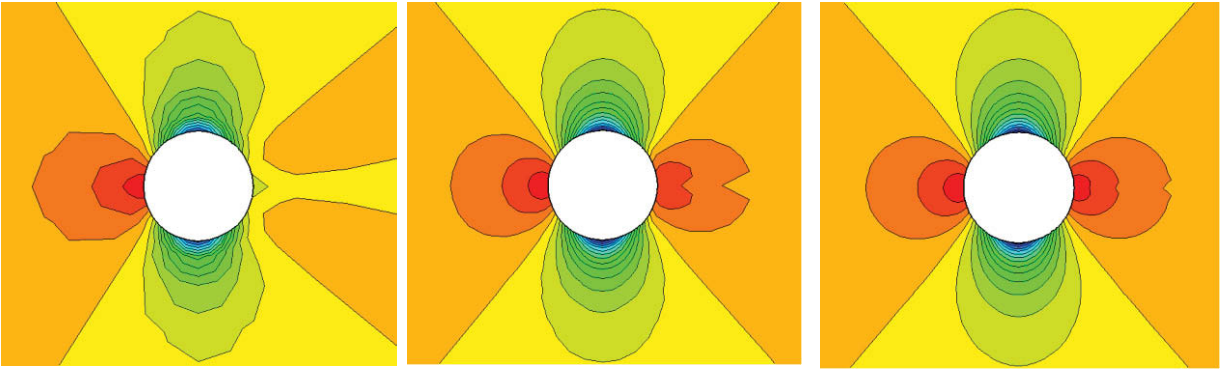


Figure 6. Computed density contours in the flow field obtained by the DG(P1) method on 32x9 mesh (left), 64x17 mesh (middle), and 128x33 mesh (right) for subsonic flow past a circular cylinder at $M_\infty=0.38$.

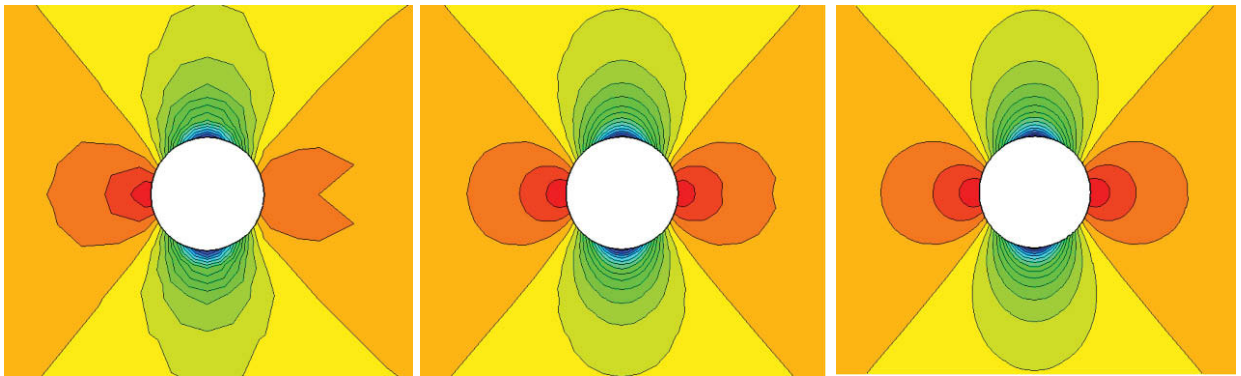


Figure 7. Computed density contours in the flow field obtained by the P1P2(RC) method on 32x9 mesh (left), 64x17 mesh (middle), and 128x33 mesh (right) for subsonic flow past a circular cylinder at $M_\infty=0.38$.

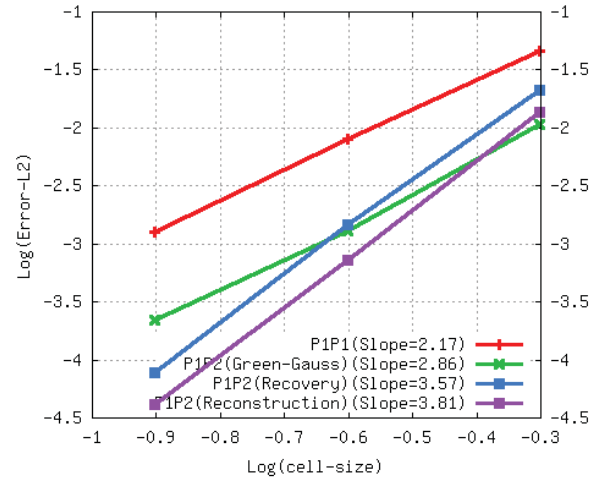


Figure 8. Convergence history for the convection of an isentropic vortex for different RDG methods.

C. Inviscid Flow through a Channel with a Smooth Bump

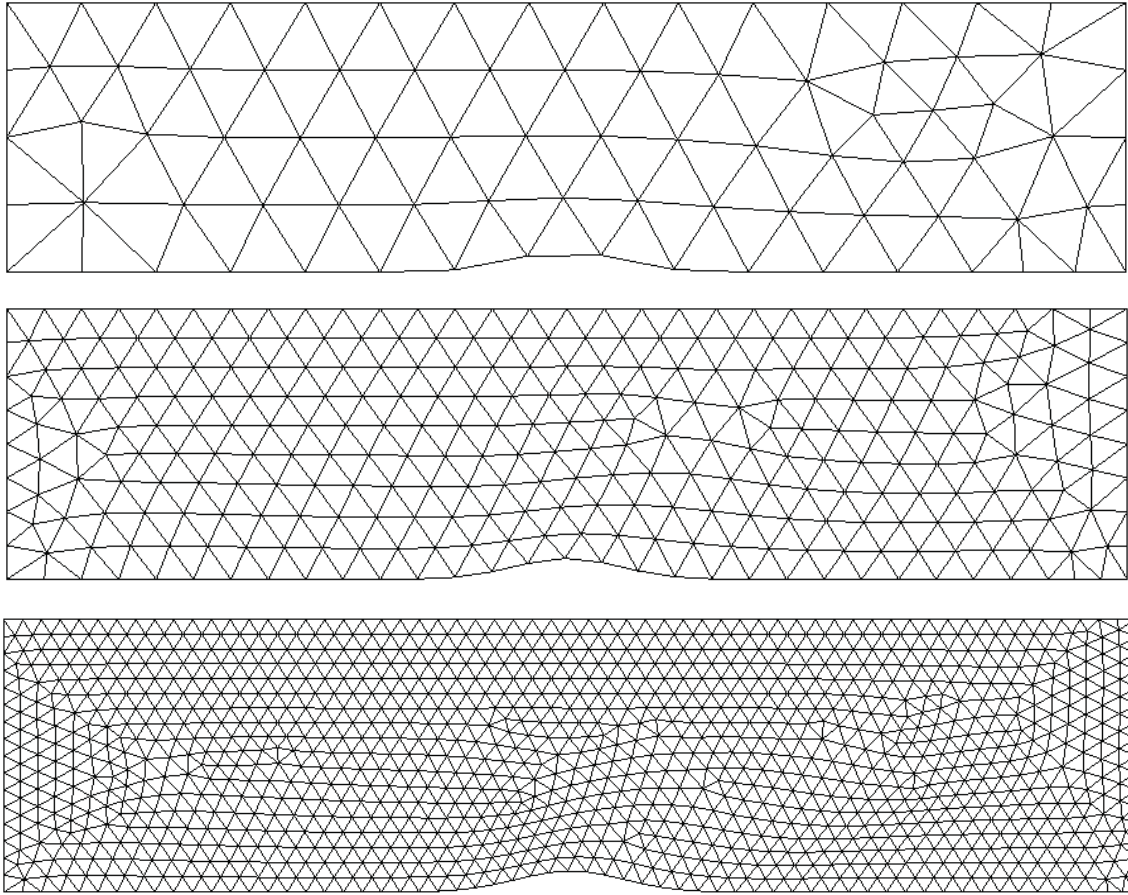


Figure 9: Sequences of three successively globally refined unstructured meshes used for computing subsonic flow in a channel with a smooth bump.

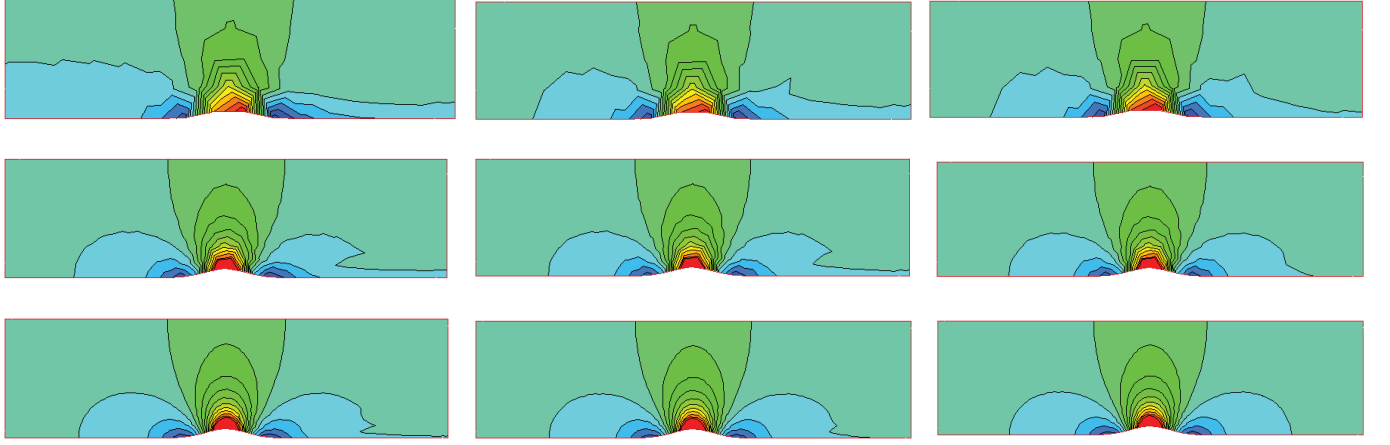


Figure 10. Computed velocity contours in the flow field obtained by the DG(P1) method (left) and least-squares reconstructed RDG(P1P2) method (right) on the coarse mesh (top), medium mesh (middle), and fine mesh (right) for a subsonic flow through a channel with a bump on the lower surface at $M_\infty=0.5$.

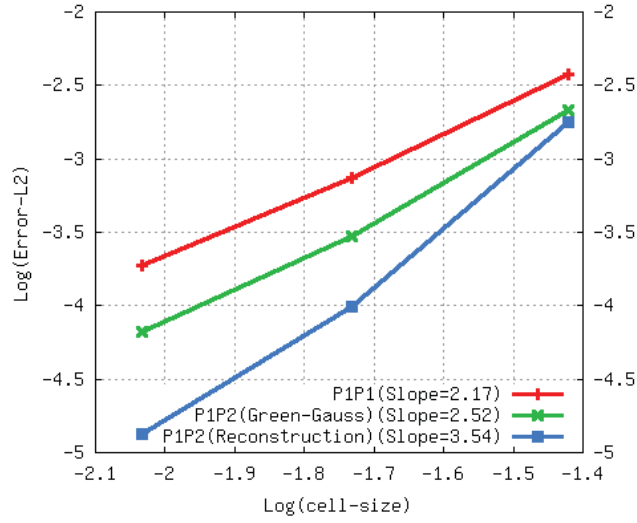


Figure 11. Convergence history for the convection of an isentropic vortex for different reconstructed RDG methods.

This test case is chosen to access the accuracy and ability of the RDG methods for computing internal flows. The problem under consideration is a subsonic flow through a channel with a smooth bump. The computational domain is bounded between -1.5 and 1.5 in the x-coordinate and the bump defined by $0.0625\exp(-25x^2)$ and 0.8 in the y-coordinate. The inflow condition is at a Mach number of 0.5, and an angle of attack of 0° . Figure 9 shows the three successively refined triangular grids used for conducting the grid convergence study, having 127, 508, and 2,032 triangles, respectively. Figure 10 illustrates the computed velocity contours in the flow field obtained by DG(P1), P1P2(GG), and P1P2(RC) methods, respectively, where one can clearly observe the significant improvement of the computed solution using the RDG method. Figure 11 provides the details of the spatial convergence of the two reconstruction-based DG methods for this numerical experiment. Again, for this internal flow problem, all reconstructed RDG(P1P2) methods offer a full $O(h^{p+2})$ order of the convergence, adding one order of accuracy to the underlying DG(P1) method. Note that the least-squares recovery DG method P1P2(rc) is unable to provide a stable solution for this test case, which is attributed to the lack of sufficient recovered equations for some of boundary cells.

D. Subsonic Flows past a NACA0012 airfoil

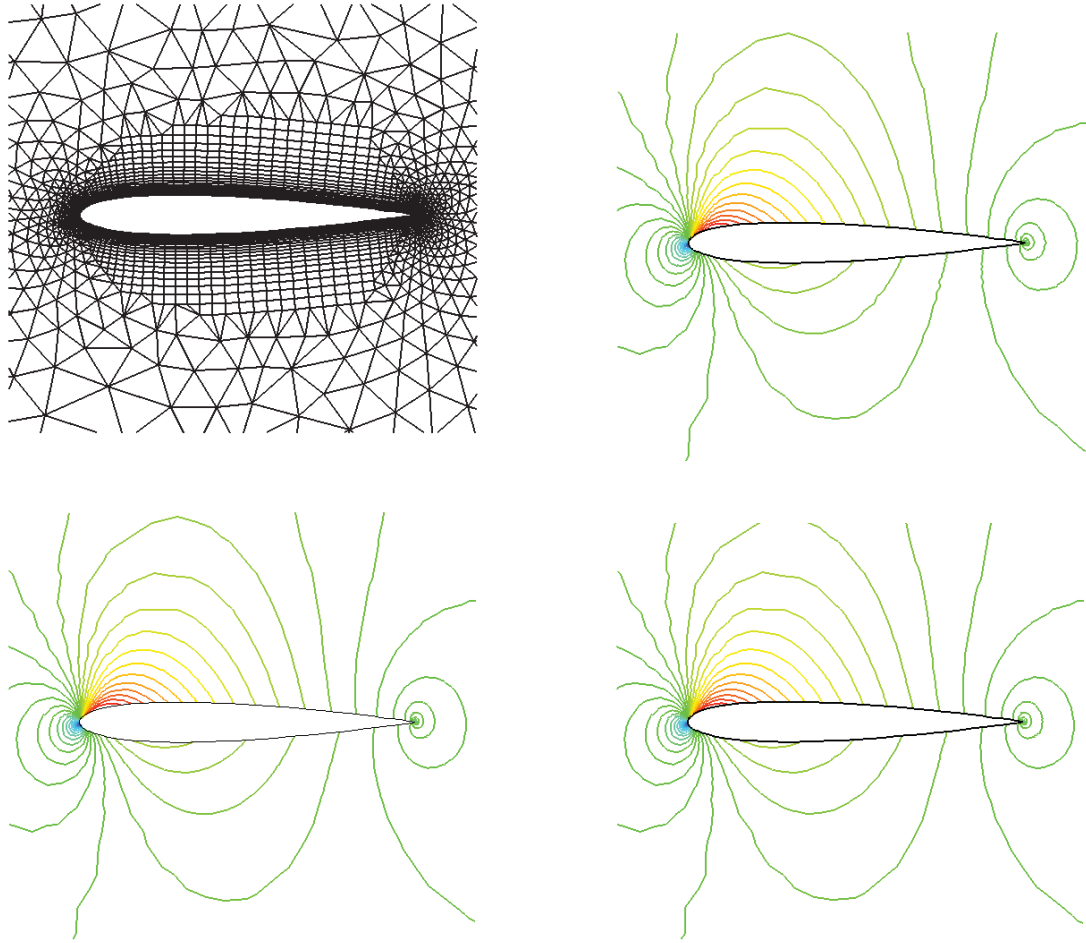


Figure 12. Unstructured hybrid mesh (top left) ($n_{quad}=1,533$, $n_{tria}=3,469$, $n_{poin}=3,346$, $n_{bfac}=157$) and computed Mach number contours by the DG(P1) (top right), the P1P1(GG) (bottom left), and the P1P2(RC)(bottom right) methods, respectively for subsonic flow past a NACA0012 airfoil at $M_{\infty} = 0.63$, $\alpha=2^{\circ}$.

Our last test case involves an inviscid flow past a NACA0012 airfoil at a Mach number of 0.63, and an angle of attack 2° . This numerical experiment is designed to test the ability of the RDG method for obtaining a highly accurate solution to the Euler equations on a viscous hybrid grid. Being able to produce a highly accurate inviscid solution on a highly stretched Navier-Stokes grid is extremely difficult and challenging, and yet of utmost importance for the accurate solution to the Navier-Stokes equations, thus serving as a good criterion to measure accuracy and robustness of a numerical method. Many finite volume methods are unable to obtain the same quality of an inviscid solution on an anisotropic Navier-Stokes grid as on an isotropic Euler grid, suffering from either excessive numerical dissipation or spurious oscillations due to a combination of a mesh irregularity and reconstruction schemes. Figure 12 shows the mesh used in this numerical experiment and the computed Mach number contours in the flow field obtained by DG(P1), P1P2(GG), and P1P2(RC) methods, respectively. The mesh consists of 1,533 quadrilateral cells, 3,469 triangular cells, 3,346 grid points, and 157 boundary faces. The computed Mach number contours in the flow field obtained using DG(P1), P1P2(GG), and P1P2(RC) methods are shown in Fig. 12, where accurate and smooth solutions are observed in spite of the highly stretched grid used in the boundary layer. The computed pressure coefficient and entropy production distribution on the surface of the airfoil obtained by these three methods are compared in Fig. 12. All three solutions are virtually identical by judging the Mach number contours in the flow field and the pressure coefficient distributions on the surface of the airfoil, indicating

that the numerical solution is order-independent, (i.e., a convergence on these flow quantities is reached). The RDG(P1) solution however is significantly improved compared with the P1P2 solution by judging the entropy production distribution on the surface of the airfoil. Note that the entropy production corresponds directly to the error of the numerical methods, as it should be zero everywhere for subsonic flows. The P1P2(RC) solution provides a further improvement over the P1P2(GG) solution, although the difference is very small. This numerical experiment demonstrates that the DG methods, unlike some of its finite volume counterparts, have the ability to accurately solve the compressible Euler equation on an anisotropic grid designed for solving the Navier-Stokes equations.

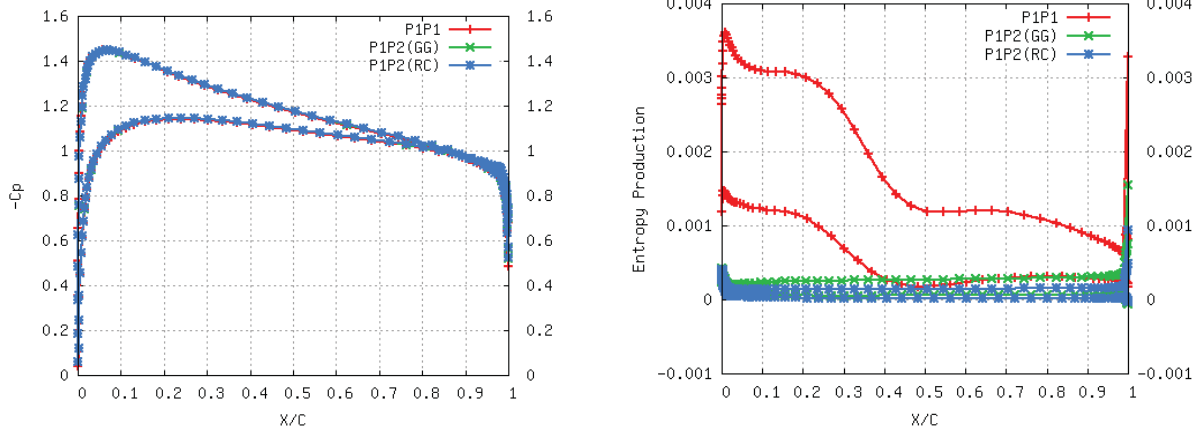


Figure 11. Comparison of the computed pressure coefficient (left) and entropy production (right) distributions for subsonic flow past a NACA0012 airfoil at $M_\infty = 0.63$, $\alpha = 2^\circ$.

E. Blasius Boundary Layer

The laminar boundary layer over an adiabatic flat plate at a free-stream Mach number of 0.2 and a Reynolds number of 100,000 based on the freestream velocity and the length of the flat plate is considered in this test case, where the computational domain is bounded from -0.5 to 1 in the x-direction and 0 to 1 in the y-direction, and the flat plate starts at point (0,0) and extends to (1,0). This problem is chosen to assess the accuracy of the different RDG methods for the solution of the compressible Navier-Stokes equations, as the Blasius solution can be used to measure accuracy of the numerical solutions. Computations are performed on two grids: one quadrilateral grid and one triangular grid, shown in Figure 12, to assess the accuracy of the reconstructed discontinuous Galerkin method on different types of grids. The quadrilateral grid used in this test case has (61x17) grid points, with 20 cells ahead of the flat plate and 40 cells for the flat plate. The stretching ratio is the ratio of the heights of the two successive elements. A stretching ratio of 1.2 is used for the quadrilateral grid in the computation. For the grid with a stretching ratio of 1.2, the height of the first element is 0.1291E-02, and the cell sizes in the x-direction for the first element at the leading and trailing edges of the flat plate are 0.12086e-02 and 0.110386, respectively. The unstructured grid consists of 900 grid points, and 105 boundary points, with 31 grid points on the flat plate. The height of the first element is 0.3464E-03 and 0.82649E-03 at the leading and trailing edge of the flat plate respectively. As a result, the unstructured grid provides the better grid resolution on the boundary layers than the quadrilateral one. The numerical results obtained by P1P2(GG), P1P2(rc), and P1P2(RC) on these two grids are presented, and compared with the theoretical one given by the well-known Blasius solution. Figure 13 shows the logarithmic plot of the computed skin friction coefficient obtained by these quadratic DG solutions on quadrilateral and triangular grids respectively. Note that the least-squares recovery DG method P1P2(rc) is unable to provide a stable solution for this test case on the triangular grid, which is attributed to the lack of sufficient recovered equations for some of boundary cells. One can observe that the least-squares reconstruction and recovery methods provide more accurate solutions for the skin friction coefficient than the Green-Gauss reconstruction method, and P1P2(RC) method outperforms P1P2(rc) method in this case, again

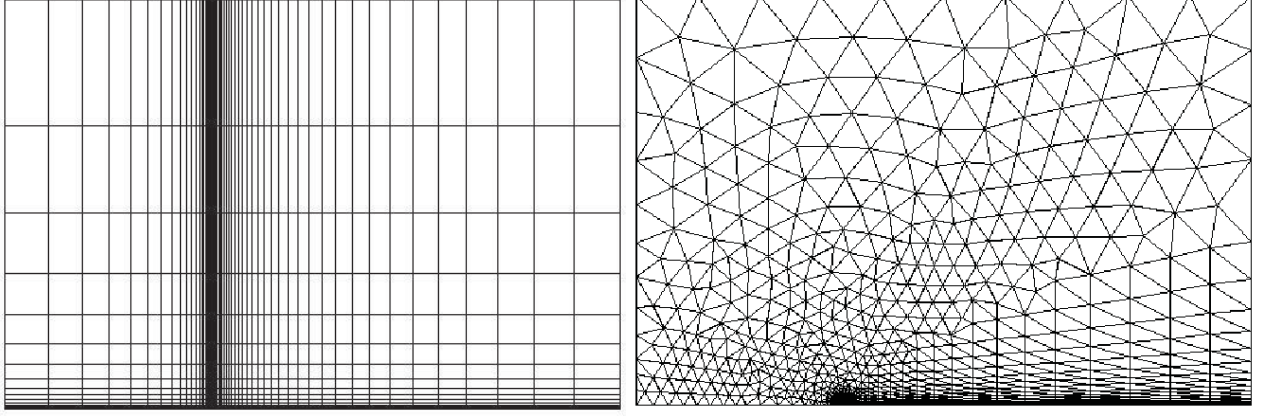


Figure 12. Quadrilateral and triangular grids used for computing a laminar flow past a flat plate at $M_\infty = 0.2$, $\alpha = 0^\circ$, $Re = 100,000$

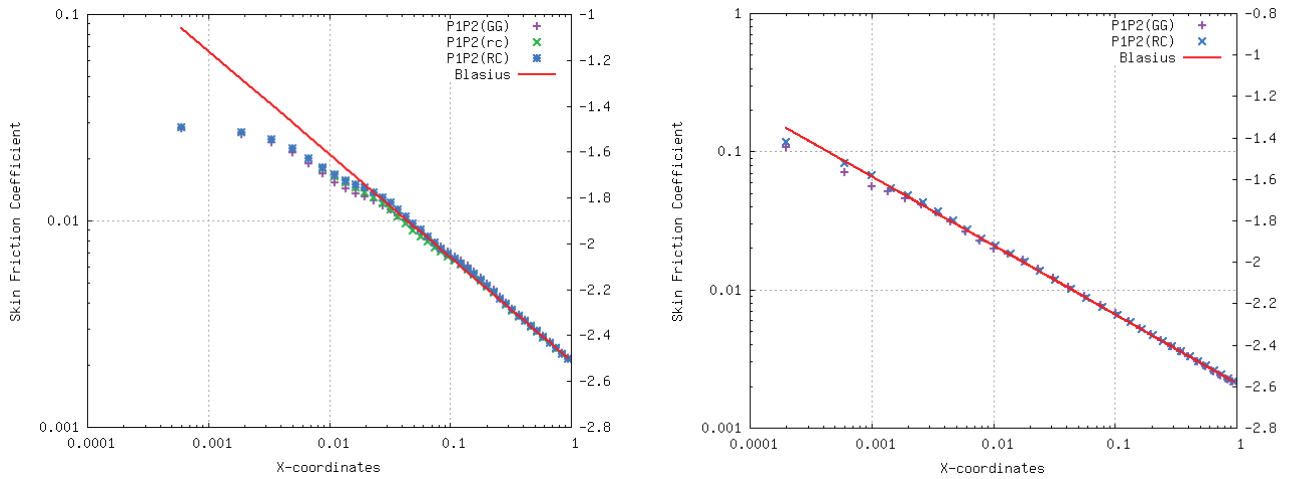


Figure 13. Logarithmic plot of the computed skin friction coefficient distribution along the flat plate obtained by the P1P2(GG), P1P2(rc), and P1P2(RC) solutions on the quadrilateral grid (left), and the triangular grid (right).

V. Conclusion

A comparative study for a class of reconstructed discontinuous Galerkin methods has been conducted for solving compressible flows on arbitrary grids. Both Green-Gauss and least-squares reconstruction methods and a least-squares recovery method have been developed to obtain a quadratic polynomial representation from an underlying linear discontinuous Galerkin solution. The developed RDG methods are compact and combine efficiency of the reconstruction methods and accuracy and robustness of the DG methods. A number of test cases have been carried out to assess the performance of these three reconstructed discontinuous Galerkin methods. The numerical experiments indicate that all three reconstructed discontinuous Galerkin methods can deliver the desired third order of accuracy and significantly improve the accuracy of the underlying second-order DG method, although the least-squares reconstruction method provides the best performance in terms of both accuracy and robustness. The future development will be focused on the extension of the reconstructed DG methods for 3D problems.

Acknowledgments

This manuscript has been authored by Battelle Energy Alliance, LLC under contract No. DE-AC07-05ID14517 (INL/CON-09-16255) with the U.S. Department of Energy. The United States Government retains and the

published, by accepting the article for publication, acknowledges that the United States Government retains a nonexclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for United States Government purposes. The authors would like to acknowledge the partial support for this work provided by DOE under Nuclear Engineering University Program. The first and last authors would also like to acknowledge the partial support for this work provided by the NSF under project No. NSF-DMS0914706.

References

- ¹Reed, W.H. Reed and T.R. Hill, "Triangular Mesh Methods for the Neutron Transport Equation," **Los Alamos Scientific Laboratory Report**, LA-UR-73-479, 1973.
- ²B. Cockburn, S. Hou, and C. W. Shu, "TV-D Runge-Kutta Local Projection Discontinuous Galerkin Finite Element Method for conservation laws IV: the Multidimensional Case," **Mathematics of Computation**, Vol. 55, pp. 545-581, 1990. eorge, P. L., Automatic Mesh Generation, J. Wiley & Sons, 1991.
- ³B. Cockburn, and C. W. Shu, "The Runge-Kutta Discontinuous Galerkin Method for conservation laws V: Multidimensional System," **Journal of Computational Physics**, Vol. 141, pp. 199-224, 1998.
- ⁴B. Cockburn, G. Karniadakis, and C. W. Shu, "The Development of Discontinuous Galerkin Method", in *Discontinuous Galerkin Methods, Theory, Computation, and Applications*, edited by B. Cockburn, G.E. Karniadakis, and C. W. Shu, Lecture Notes in Computational Science and Engineering, Springer-Verlag, New York, 2000, Vol. 11 pp. 5-50, 2000.
- ⁵F. Bassi and S. Rebay, "High-Order Accurate Discontinuous Finite Element Solution of the 2D Euler Equations," **Journal of Computational Physics**, Vol. 138, pp. 251-285, 1997.
- ⁶H. L. Atkins and C. W. Shu, "Quadrature Free Implementation of Discontinuous Galerkin Method for Hyperbolic Equations," **AIAA Journal**, Vol. 36, No. 5, 1998.
- ⁷F. Bassi and S. Rebay, "GMRES discontinuous Galerkin solution of the Compressible Navier-Stokes Equations," **Discontinuous Galerkin Methods, Theory, Computation, and Applications**, edited by B. Cockburn, G.E. Karniadakis, and C. W. Shu, Lecture Notes in Computational Science and Engineering, Springer-Verlag, New York, 2000, Vol. 11 pp. 197-208, 2000.
- ⁸T. C. Warburton, and G. E. Karniadakis, "A Discontinuous Galerkin Method for the Viscous MHD Equations," **Journal of Computational Physics**, Vol. 152, pp. 608-641, 1999.
- ⁹J. S. Hesthaven and T. Warburton, "Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications," Texts in **Applied Mathematics**, Vol. 56, 2008.
- ¹⁰P. Rasetarinera and M. Y. Hussaini, "An Efficient Implicit Discontinuous Spectral Galerkin Method," **Journal of Computational Physics**, Vol. 172, pp. 718-738, 2001.
- ¹¹B. T. Helenbrook, D. Mavriplis, and H. L. Atkins, "Analysis of p -Multigrid for Continuous and Discontinuous Finite Element Discretizations," **AIAA Paper** 2003-3989, 2003.
- ¹²K. J. Fidkowski, T. A. Oliver, J. Lu, and D. L. Darmofal, " p -Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations," **Journal of Computational Physics**, Vol. 207, No. 1, pp. 92-113, 2005.
- ¹³H. Luo, J. D. Baum, and R. Löhner, "A Discontinuous Galerkin Method Using Taylor Basis for Compressible Flows on Arbitrary Grids," **Journal of Computational Physics**, DOI: 210.1016/j.jcp.2008.06.035, Vol. 227, No 20, pp. 8875-8893, October 2008.
- ¹⁴H. Luo, J.D. Baum, and R. Löhner, "On the Computation of Steady-State Compressible Flows Using a Discontinuous Galerkin Method", **International Journal for Numerical Methods in Engineering**, Vol. 73, No. 5, pp. 597-623, 2008.
- ¹⁵H. Luo, J. D. Baum, and R. Löhner, "A Hermite WENO-based Limiter for Discontinuous Galerkin Method on Unstructured Grids," **Journal of Computational Physics**, Vol. 225, No. 1, pp. 686-713, 2007.
- ¹⁶H. Luo, J.D. Baum, and R. Löhner, "A p -Multigrid Discontinuous Galerkin Method for the Euler Equations on Unstructured Grids", **Journal of Computational Physics**, Vol. 211, No. 2, pp. 767-783, 2006.
- ¹⁷H. Luo, J.D. Baum, and R. Löhner, "A Fast, p -Multigrid Discontinuous Galerkin Method for Compressible Flows at All Speeds", **AIAA Journal**, Vol. 46, No. 3, pp.635-652, 2008.
- ¹⁸M. Dumbser, D.S. Balsara, E.F. Toro, C.D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. **Journal of Computational Physics**, 227:8209-8253, 2008.
- ¹⁹M. Dumbser, O. Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. **Journal of Computational Physics**, 228:6991-7006, 2009.
- ²⁰M. Dumbser. Arbitrary High Order PNPM Schemes on Unstructured Meshes for the Compressible Navier-Stokes Equations. **Computers & Fluids**, 39: 60-76. 2010.
- ²¹F. Bassi and S. Rebay, "A High-Order Accurate Discontinuous Finite Element Method for the Numerical Solution of the Compressible Navier-Stokes Equations," **Journal of Computational Physics**, Vol. 131, pp. 267-279, 1997.
- ²²F. Bassi and S. Rebay, "Discontinuous Galerkin Solution of the Reynolds-Averaged Navier-Stokes and $k-\omega$ Turbulence Model Equations," **Journal of Computational Physics**, Vol. 34, pp. 507-540, 2005.
- ²³B. Cockburn and C.W. Shu, "The Local Discontinuous Galerkin Method for Time-dependent Convection-Diffusion System," **SIAM, Journal of Numerical Analysis**, Vo. 16, 2001.

- ²⁴C. E. Baumann and J. T. Oden, "A Discontinuous hp Finite Element Method for the Euler and Navier-Stokes Equations," **International Journal for Numerical Methods in Fluids**, Vol. 31, 1999.
- ²⁵J. Peraire and P. O. Persson, The Compact Discontinuous Galerkin Method for Elliptic Problems. **SIAM Journal on Scientific Computing**, 30: 1806-1824, 2008.
- ²⁶D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems", **SIAM Journal on Numerical Analysis**. Vol. 39, No. 5., pp. 1749-1779, 2002.
- ²⁷G. Gassner, F. Lorcher, and C. D. Munz, "A Contribution to the Construction of Diffusion Fluxes for Finite Volume and Discontinuous Galerkin Schemes", **Journal of Computational Physics**, Vol. 224, No. 2, pp. 1049-1063, 2007.
- ²⁸H. Liu and K. Xu, "A Runge-Kutta Discontinuous Galerkin Method for Viscous Flow Equations", **Journal of Computational Physics**, Vol. 224, No. 2, pp. 1223-1242, 2007.
- ²⁹H. Luo, L. Luo and K. Xu - A Discontinuous Galerkin Method Based on a BGK Scheme for the Navier-Stokes Equations on Arbitrary Grids, **Advances in Applied Mathematics and Mechanics**, Vol. 1, No. 3, pp. 301-318, 2009.
- ³⁰B. van Leer and S. Nomura, "Discontinuous Galerkin Method for Diffusion," AIAA Paper 2005-5108, 2005.
- ³¹B. van Leer and M. Lo, "A Discontinuous Galerkin Method for Diffusion Based on Recovery," AIAA Paper 2007-4083, 2007.
- ³²M. Raalte and B. van Leer, "Bilinear Forms for the Recovery-Based Discontinuous Galerkin Method for Diffusion," **Communication of Computational Physics**, Vol. 5, No. 2-4, pp. 683-693, 2009.
- ³³R. Nourgaliev, H. Park, and V. Mousseau, "Recovery Discontinuous Galerkin Jacobian-Free Newton-Krylov Method for Multiphysics Problems," **Computational Fluid Dynamics Review** 2010, 2010.
- ³⁴H.T Huynh, "A Reconstruction Approach to High-order Schemes Including Discontinuous Galerkin for Diffusion," AIAA Paper 2009-0403, 2009.
- ³⁵H. Luo, L. Luo, R. Nourgaliev, and V. Mousseau, A Reconstructed Discontinuous Galerkin Method for the Compressible Euler Equations on Arbitrary Grids, AIAA-2009-3788, 2009.
- ³⁶H. Luo, L. Luo, R. Nourgaliev, Mousseau, A, and N. Dinh, A Reconstructed Discontinuous Galerkin Method for the Compressible Navier-STokes Equations on Arbitrary Grids, **Journal of Computational Physics**, Vol. 229, pp. 6961-6978, 2010.
- ³⁷H. Luo, Luo, L., Ali, A., Norgaliev, R., and Cai, C., A Parallel, Reconstructed Discontinuous Galerkin Method for the Compressible Flows on Arbitrary Grids, **Communication in Computational Physics**, Vo. 9, No.2, pp. 363-389, 2011.
- ³⁸Zhang L.P, Liu W., He L.X., Deng, X.G., Zhang H.X., A Class of Hybrid DG/FV Schemes for 2D Conservation Scalar Law, The Sixth International Conference on Computational Fluid Dynamics, St. Petersburg, Russia, July, 12-16, 2010.
- ³⁹Zhang L.P, Liu W., He L.X., Deng, X.G., Zhang H.X., A Class of Hybrid DG/FV Methods for Conservation Laws II: Two-dimensional Cases, To be appeared in **Journal of Computational Physics**.